DRAWING AMENDMENTS

Please replace the originally filed Figure 2 with the enclosed amended version of Figure 2. The amended Figure 2 has the following alterations relative to the originally filed figure:

The module BUFFER shown within the I/O CTRL module 470 has been eliminated, along with the reference number and lead line.

The module TRANSFORM has been renumbered from 476 to 474.

REMARKS

Pending claims

Assuming entry of this amendment, claims 1-32 are still pending.

Specification Amendments

Most of the specification changes are requested to correct the grammar error of including an apostrophe in the plural form of the abbreviations VM and VMM. This of course adds no new matter.

The change (other than the correction of a typographical error) requested in paragraph [0028] of the specification, however, relates to the definition of the virtual machine monitor. As stated in the last sentence, "[t]he general features of VMMs are known in the art and are therefore not discussed in detail here." Thus, this paragraph is summarizing known features of virtual machine monitors. One known feature of modern VMMs is that it is not necessary (although frequently implemented) for a VMM to virtualize all the hardware resources of the hardware platform on which it runs; rather, it may virtualize only a subset, or, indeed, a superset, of the resources, and these need not relate to the actual hardware present. The change to the specification is requested in order to reflect this known property; it also adds no new matter, but rather simply better describes known VMMs.

Drawing Amendments

In paragraph [0044] it is stated (emphasis added):

> Data ... that is to be transferred to (or from) the VM ... is stored in a memory space or module 472 referred to here as the **"map"** or **"buffer"** or "mask" depending on what type of information it is currently being used to store. For example, as is explained below in greater detail, this module may be a bit map used for a display, or it may be a buffer space where data transferred serially via a network is first assembled before it is transformed.

The modules MAP and BUFFER shown in Figure 2 are therefore the same thing, such that the BUFFER module is unnecessary. Throughout the description, the transformation module is numbered 474, and the number 476 does not appear at all.

The requested changes to Figure 2 are therefore intended to make sure that th efigure conforms to the text, and add no new matter.

Claim Rejections

Rejections Under 35 U.S.C. § 112

The Examiner rejected claim 25 because of lack of antecedent basis for "the step of filtering" and "the transformation-triggering criterion." As for the filter step, the problem was with claim dependence: Claim 25 should have referred back to claim 3, not claim 1, since claim 3 provides the antecedent basis for this feature. As for the criterion, the indefinite article "a" now replaces the previous definite article "the".

The Examiner rejected claims 8 and 26 for indefiniteness. In claim 8, the Examiner observed that the phrase "in the presence in the I/O data of a copy protection indication" appeared to have a grammatical error of some sort. The problem was typographical: The correct phrase should be "*is* the presence in the I/O data of a copy protection indication." Thus (after all requested amendments) claim 8 now states: "… in which the filtering condition is the presence in the I/O data of a copy protection indication."

As for claim 26, the rejection was based on this claim's dependence on the previously indefinite claim 25. Since claim 25 has now been "fixed," claim 26 should now be definite.

Rejections Under 35 U.S.C. § 102

The Examiner rejected claims 1-3, 9-11 and 25-32 under 35 U.S.C. 102(e) as being anticipated by US 2002/0143842 ("Cota-Robles"). Of these, claims 1, 27 and 28 are independent base claims for all the other pending claims.

Concerning claim 1, the Examiner wrote that, among other features of claim 1, Cota-Robles discloses:

> performing a predetermined transformation of I/O data passing between
> the VM and the device (paragraphs 0015, 0027, 0047);
> the transformation of th eI/O data thereby being undefeatable by any user
> action via the VM (paragraphs 0025, 0027, 0029, 0047).

Paragraphs [0024]- [0027] and [0029] of Cota-Robles perhaps best bring out the

fundamental difference between Cota-Robles and the invention (emphasis added):

[0024] In some embodiments (e.g., when the corresponding functionality is performed by the host processor), the soft device driver 208 emulates the functionality of a fixed function device by performing *the requested computations* without any notification to the residual fixed function device 204.

[0025] In other embodiments, the soft device driver may pass the read or write command without alteration to the residual fixed function device 204 *to perform an operation requested by the guest OS* 106 or 114.

[0026] In still other embodiments, the soft device driver 208 may perform some of the requested computations and may then perform one or a whole series of similar or completely different operations on the actual hardware registers of the residual fixed function device 204 *in order to complete the task that the guest OS 106 or 114 requested.*

[0027] In yet other embodiments, the residual fixed function device 204 is capable of directly accessing system memory. In one embodiment, the soft device driver 208 performs zero or more transformations on data received from guest OS 106 or 114 before directly transferring the data to system memory at a predefined location from which the residual fixed function device 204 will fetch the data *in order to effect the operation requested by the guest OS* 106 or 114. In another embodiment, the soft device driver 208 performs zero or more transformations on data directly transferred to system memory at a predefined location by the residual fixed function device 204, said transformed data then being transferred to the guest OS 106 or 114 *in order to complete the requested operation.*

[0029] Subsequently, when a request to use the soft device is received from a virtual machine coupled to the VMM (processing block 306), the soft device is made available for use by this virtual machine (processing block 308). In one embodiment, making the soft device available for use by the virtual machine includes … emulating the functionality of a fixed function hardware device *to perform the read and or write operation requested by the virtual machine*. In one embodiment, this functionality is emulated by the soft device driver performing the requested computations without any notification to a hardware component of the soft device (i.e., a residual fixed function device). In an

alternative embodiment, the functionality is emulated by the hardware component of the soft device which receives the read or write command from the soft device driver and executes this command **to perform an operation requested by the virtual machine.** In still another embodiment, the soft device driver performs some of the requested computations and then performs one or a whole series of similar or completely different operations on the actual hardware registers of the residual fixed function device **in order to complete the task intended by the virtual machine.** In yet another embodiment, the residual fixed function device 204 is capable of directly accessing system memory and transfers data to or from system memory at predefined locations, and the soft device driver manipulates data stored in memory **according to the operation requested by the virtual machine.** ...For instance, the soft device driver performs the write command received from the virtual machine by transforming the data in the memory appropriately and then transferring the requested data to the memory using the direct memory access (DMA) technique.

In general, the transformations that Cota-Robles' system performs are intended to enable commands made to one device (e.g. the soft device) to be made compatible with commands to another device (e.g. the hardware device). Consequently, as explained in the many highlighted portions of paragraphs of Cota-Robles above, Cota-Robles transforms data specifically to be able to complete a request by the VM. In other words, such transformations as Cota-Robles even performs are so that the soft device driver will be able to complete the VM requests at all. In other words, Cota-Robles' transformations are steps required to complete the data transfer request as issued by the VM, that is, as the VM intends the transfer to be completed.

In contrast, the transformations performed by the invention are separate from any operation that is required to complete the I/O request as such, that is, as issued by the requesting VM. In fact, as is mentioned in the specification, not only are the transformations according to the invention not part of any request as such made by the VM, but they may also not even be something the VM user wants. On-screen banners and tags are two examples. See, for example, paragraph [0054] of the specification:

> The result of this transformation operation is that the banner or tag or other visible indication 920 will appear on the physical display regardless of what the user does -- because the user's actions are restricted to the VM, he has no way to influence the actual display map 472 used to generate the physical display. In other words, the user 800 cannot defeat (prevent or undo

or in any way affect) the transformation of the requested I/O operation even if he somehow were able to gain direct control of the VOS 320 and all other components of the VM: untransformed data is simply never available for access by the physical display driver because the VMM is transparent to the VM.

Many other examples are described in the specification.

Mindful of the prohibition against claiming in the negative (what an invention does *not* do), the applicants have therefore amended claims 1 and 28 such that the relevant claim element now reads: "performing a transformation of I/O data passing between the VM and the device, said transformation being adjunct to necessary completion of the request, as issued, for the I/O operation"

As Webster's New World College Dictionary, Fourth Edition, Wiley Publishing, 2002, explains, "adjunct" means: "a thing added to something else, but secondary or not essential to it." Thus, as claim 1 now recites, the transformation performed by the invention is *adjunct* to the necessary completion of the I/O request as issued. In other words, from the perspective of the VM, the transformations are therefore something extra – something not a part of the assumed steps needed for completion of the request.

Claim 27 has been similarly amended to recite: "said replacement data being entered as a processing step that is adjunct to the necessary completion of the I/O operation".

This key feature of the invention is therefore not only not taught by Cota-Robles, but is in fact the *opposite* of what is taught in Cota-Robles. As such, claims 1, 27 and 28 should be allowable over Cota-Robles in that they recite a feature that is not found in Cota-Robles, and that provides a different outcome of an I/O request, with a unique advantage.

Rejections Under 35 U.S.C. § 103(a)

The Examiner rejected the remaining claims as being obvious in view of hypothetical combinations of Cota-Robles and various other secondary references. By definition, these claims, all of which were dependent on claim 1, 27 or 28, include the limitations of their respective base claims. As such, *all* of the pending claims now either explicitly recite or by definition incorporate the limitation that the transformation performed in response to the requested I/O operation is adjunct to the necessary completion of the operation. As explained above, Cota-Robles fails to teach this, and in fact teaches the opposite. None of the cited secondary references teach this feature either. Consequently, no combination of Cota-Robles and any of the secondary references teach the invention as now defined in the independent claims. Accordingly, the applicants respectfully submit that all of the claims now distinguish the invention over the cited prior art, and should be allowable.

Voluntary Claim Amendments

The term "predetermined" has been eliminated from the claims, in order to avoid confusion: The I/O transformation is predetermined in the sense that, in most cases, it will be known that the data is to be encypted, or provided with a display overlay, etc., (the specification lists many other possibilities), because this will be encoded in the transformation module 474, but this does not always mean that the exact result of the transformation will be predetermined, since the data input to the transformation module will usually not be known in advance. The word "predetermined" is also superfluous to the claims.

In claims 1, 27 and 28, the last claim element has been amended thus: "the transformation of the I/O data thereby being undefeatable by any ~~user~~ action <u>initiated</u> via the VM." This is because it is possible that the guest ("virtual") operating system might, for example, cause an I/O request to be initiated without any specific user action. The important point is simply that something within the VM initiates the request, be it according to direct user input or according to some routine within the guest operating system or some other software entity within the VM.

Conclusion

The various embodiments of the applicant's invention as defined in the corresponding independent claims recite features that are not found at all in either of the cited references, whether the references are viewed independently or in combination. As such, the independent claims should now be allowable over the cited prior art. The various dependent claims of course simply add additional limitations and should therefore be allowable along with their respective independent base claims.

Change of Attorney Name

Please note the enclosed letter concerning the change of the attorney's name from "Slusher" to "Pearce."

Date: 16 March 2005

Respectfully submitted,

Jeffrey Pearce

VMware, Inc.
ATTN: Jeffrey Pearce
3145 Porter Drive
Palo Alto, CA 94304
Phone & fax: 360-793-6687

Jeffrey Pearce
Reg. No. 34,729
Attorney for the Applicant